A Highly Generic Real-time Monitoring and Display System

F. Hervias
U. M. Schwuttke

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109

email: felipe@puente.jpl.nasa.gov

Extended Abstract

There are numerous definitions of real-time systems, the most stringent of which involve guaranteeing correct system response within a domain-dependent or situation-defined period of time. However, another definition of real-time is relevant in the case of applications where failure to supply a response in the proper (and often infinitesimal) amount of time allowed does ***not*** make the solution less useful (or, in extreme examples, completely irrelevant), This more casual definition involves responding to data at the same rate at which it is produced, and is most appropriate for applications with softer real-time constraints, such as the monitoring of inter-planetary spacecraft, which results in massive quantities of data that travel at the speed of light for a number of hours before they even reach the monitoring system.

This is the definition of real-time applied to a generic and interactive monitoring and display system that is being developed to support interplanetary mission operations at NASA's Jet Propulsion Laboratory. The current system is a third generation development that builds on lessons learned in previous advanced development efforts at JPL. This system is being applied to the monitoring of health and status data from satellites or interplanetary spacecraft, although it is sufficiently generic to be suitable for other, unrelated kinds of monitoring applications. The application environment is one that is characterized by complex human interactions, large quantities of data, and a combination of challenging and mundane engineering activities.

Background

Up through the final Voyager encounters, spacecraft monitoring was a manually intensive task that involved dedicated teams of experts for each spacecraft subsystem. Each team was responsible for manually verifying all the telemetry received from its spacecraft subsystem to assure that the data was within expected ranges or corresponded to predicted values.This required the presence of approximately ten people around-the-clock in order to support real-time monitoring activity, and closer to 100 people to provide more in-depth non-real-time support. More recent missions like Galileo (a Jupiter orbiter and probe) and TOPEX (an earth-orbiting mission studying ocean topography) have begun to benefit from software that automatically monitors telemetry and displays the results in real-time. There have been several operational prototypes that have successfully included traditional non-real-time functions such as trend analysis, automated report generation, and embedded expert systems for providing automated diagnosis in order to achieve work force reductions.

Detailed analysis of these operational prototypes has resulted in a wealth of lessons learned from both performance and development perspectives. Analysis of performance has provided insight into how to best develop software that meets our real-time definition in the UNIX/MOTIF environment, including specific insights on user interface development. Analysis of our software development process has led to the conclusion that a disproporlionate amount of effort is spent in developing highly custom, special purpose displays in order to provide end-users with maximum information content in an optimal performance system. The new system is being implemented as a highly generic, user-customizable system with implementation trade-offs being made to balance development efficiency and end-user flexibility while providing equal or better performance to that obtained from previous monitoring systems.

Requirements

The following high-level requirements exist for this application:

- Compatibility with the existing operations environment
- Compliance with existing hardware/software standards (HP or Sun Workstations, X Windows, MOTIF, etc.)
- User-definable organization (or hierarchical grouping) of data to be monitored
- User-definable data display formats for each grouping (i.e., tabular, graph, list)
- Automatic detection of out-of-range values
- Automatic detection of values that contradict prediction
- Trend analysis and automatic report generation
- Ability to monitor and display a minimum of 2000 parameters per second, while simultaneously accommodating interactive user-requests within a “reasonable” response time (currently defined as keyboard/mouse response within < 1 second).

Design

System performance and user flexibility led to a tree-structured user interface window design approach for establishing relationships between data categories and corresponding data displays. Data categories are represented as a single tree node called a Panel window. A Panel can access and bring into view other panels and/or data displays. Panels are user-customizable, and new child-nodes (i.e., other panels or data displays) can be added, updated, or removed.

Displays contain data values that can be viewed in tabular, graph, or list format. Each data display is attached to one or more panel(s) from which it earl be accessed. Newdatadisptays can be created and updated with any number of data parameters and attached to a panel. For each available display format available there is a wide range of configuration options, including data sampling, out-of-range value checking, predict verification, archival, etc. Providing user-customization of the displays permits the data to be optimally represented to end-users.

Automatic monitoring shows another advantage of this system, When an alarm condition is triggered (i.e., out-of-range value or violation of prediction), the corresponding data display will denote the alarm state. The alarm condition will be propagated via the panel tree structure until the top panel indicates the alarm condition, The advantage of this design is that by viewing the top-level panel an analyst can monitor vast amounts of data and quickly access the data display with the alarm condition simply by traversing the panel tree structure,

All the user-customizable features can be stored using four types of configuration files: panel setup, data display setup, out-of-range setup, and predict setup. The idea is to allow for combinations of files to be used interchangeably.

The design provides a flexible, generic environment for analysts from different missions to monitor and display data.